

A differential evolution based algorithm for constrained multiobjective structural optimization problems

D.E.C. Vargas¹², A.C.C. Lemonge³, H.J.C. Barbosa⁴⁵, H.S. Bernardino⁴

1 Programa de Pós-Graduação em Modelagem Computacional, Universidade Federal de Juiz de Fora, Juiz de Fora, Brasil

2 Instituto Federal do Sudeste de Minas Gerais, campus Rio Pomba, Brasil

3 Departamento de Mecânica Aplicada e Computacional, Faculdade de Engenharia, Universidade Federal de Juiz de Fora, Juiz de Fora, Brasil

4 Departamento de Ciência da Computação, Instituto de Ciências Exatas, Universidade Federal de Juiz de Fora, Juiz de Fora, Brasil

5 Laboratório Nacional de Computação Científica, Petrópolis, Brasil

Abstract

Structural optimization problems aim at increasing the performance of the structure while decreasing its costs guaranteeing, however, the applicable safety requirements. As these aspects are conflicting, the formulation of the structural optimization problem as multiobjective is natural but uncommon, and has the advantage of presenting a diverse set of solutions to the decision maker(s). The literature shows that Evolutionary Algorithms (EAs) are effective to obtain solutions in multiobjective optimization problems, and that the Differential Evolution (DE) based ones are efficient when solving structural mono-objective structural optimization problems, specially those with a real encoding of the design variables. On the other hand, one can note that DE has not yet been applied to the multiobjective version of these problems. This article presents a performance analysis of a DE-based algorithm in five multiobjective structural optimization problems. The obtained results are compared to those found in the literature, and the comparisons indicate the potential of the proposed algorithm.

OPEN ACCESS

Published: 01/06/2016

Accepted: 09/02/2015

Submitted: 15/10/2014

DOI:
10.1016/j.rimni.2015.02.003

Keywords:
Structural Optimization
Differential Evolution
Multiobjective Optimization

D.E.C. Vargas^{a, e, *}, A.C.C. Lemonge^b, H.J.C. Barbosa^{c, d}, H.S. Bernardino^c

^aPrograma de Pós-Graduação em Modelagem Computacional, Universidade Federal de Juiz de Fora, Juiz de Fora, Brasil

^bDepartamento de Mecânica Aplicada e Computacional, Faculdade de Engenharia, Universidade Federal de Juiz de Fora, Juiz de Fora, Brasil

^cDepartamento de Ciência da Computação, Instituto de Ciências Exatas, Universidade Federal de Juiz de Fora, Juiz de Fora, Brasil

^dLaboratório Nacional de Computação Científica, Petrópolis, Brasil

^eInstituto Federal do Sudeste de Minas Gerais, campus Rio Pomba, Brasil

Under a Creative Commons [license](#)

Resumo

Problemas de otimização estrutural visam o aumento do desempenho da estrutura e a diminuição de seus custos garantindo, entretanto, os requisitos de segurança aplicáveis. Devido à natureza conflitante desses aspectos, a formulação de um problema de otimização estrutural como multiobjetivo é

natural, embora pouco frequente, e tem a vantagem de apresentar um conjunto diversificado de soluções ao(s) tomador(es) de decisão. A literatura mostra que os algoritmos evolucionários (AE) são eficazes na obtenção de soluções em problemas de otimização multiobjetivo e que aqueles baseados em evolução diferencial (ED) são eficientes na resolução de problemas de otimização estrutural mono-objetivo, especialmente os que utilizam codificação real em suas variáveis de projeto. Por outro lado, nota-se a ausência da aplicação da ED na versão multiobjetivo desses problemas. Esse artigo apresenta uma análise do desempenho de um algoritmo baseado em ED em cinco exemplos de problemas de otimização estrutural multiobjetivo. Os resultados obtidos são comparados aos encontrados na literatura, indicando o potencial do algoritmo proposto.

Palavras-chave

Otimização estrutural; Evolução diferencial; Otimização multiobjetivo

1. Introdução

A solução de problemas de otimização estrutural (OE) é de grande aplicabilidade na busca por concepções estruturais com baixo custo, alto desempenho, de fácil execução e manutenção e, também, mais recentemente, incorporando aspectos ambientais, desde sua construção até à sua utilização. Na sua quase totalidade, estes problemas apresentam restrições de

várias naturezas que devem ser satisfeitas para que se obtenha uma solução candidata viável.

Em geral, os objetivos desejados são conflitantes, como a minimização de custo e a manutenção das condições mínimas de segurança, caracterizando assim um problema multiobjetivo. Devido a essa natureza conflitante, a resolução de um problema multiobjetivo obtém um conjunto diversificado de soluções de onde o(s) tomador(es) de decisão tem a oportunidade de selecionar aquela que melhor se adapta às suas preferências.

Entre as várias técnicas para o tratamento de problemas multiobjetivo estão os algoritmos evolucionários (AE). Os AEs são considerados interessantes na obtenção de soluções desses problemas por serem capazes de lidar com características normalmente encontradas neles, tais como não-linearidade e não-diferenciabilidade [1].

Um exemplo frequente de problema de OE multiobjetivo consiste em encontrar as áreas das seções transversais das barras (variáveis de projeto) de uma treliça buscando minimizar simultaneamente o seu peso e o deslocamento máximo dos seus nós. Neste caso, as restrições podem ser as tensões normais máximas nas barras.

Em Angelo et al. [2] são discutidos 5 problemas desse tipo, comumente encontrados na literatura tratados como mono-objetivo. Naquele artigo os autores utilizaram 2 algoritmos baseados em colônia de formigas: *Multi-objective Ant System* (MOAS) e *Multi-objective Ant Colony System* (MOACS). Para o tratamento das restrições adotou-se a técnica de penalização adaptativa (APM), proposta por Barbosa e Lemonge [3], cuja principal vantagem é a ausência de parâmetros a serem definidos pelo usuário. Os resultados apresentados mostraram que os algoritmos MOAS e MOACS foram capazes de gerar uma grande variedade de soluções, algumas delas muito próximas das soluções da versão mono-objetivo encontradas na literatura, sugerindo o bom desempenho dos algoritmos propostos.

Versões mono-objetivo de 4 desses 5 problemas foram testadas por Silva et al. [4] para avaliar o desempenho de um AE baseado em evolução diferencial (ED) em problemas desse tipo. Os autores também utilizaram a técnica APM para o tratamento das restrições e codificação real para suas variáveis de projeto. Os resultados apresentados mostram a potencialidade do uso da ED nesses problemas, pois quando não foram melhores mostraram-se competitivos com os encontrados na literatura.

Um levantamento das pesquisas existentes sobre AE aplicados a problemas de OE multiobjetivo é apresentado por Zavala et al. [1]. Os autores não encontraram publicações utilizando AE baseados em ED para a obtenção de soluções de problemas de OE multiobjetivo. Eles afirmam que o surgimento de trabalhos nessa direção é esperado nos próximos anos visto que a maioria dos estudos analisados utilizavam codificação real para suas variáveis de projeto e a ED tem uma potencialidade com esse tipo de codificação. O algoritmo *The Third Step of Generalized Differential Evolution* (GDE3), proposto por Kukkonen e Lampinen [5], é citado pelos autores como um algoritmo que pode ser utilizado nesses próximos trabalhos.

Baseado na argumentação apresentada e na sugestão de Zavala et al. [1], o presente trabalho analisou o desempenho do algoritmo GDE3 nos 5 problemas de OE de treliças tratados no trabalho de Angelo et al. [2]. Os resultados foram comparados aos obtidos pelos algoritmos MOAS e MOACS nas mesmas métricas de análise de desempenho utilizadas por esses autores.

Apesar do GDE3 já possuir um tratamento de restrições próprio,

a mesma técnica APM usada em Angelo et al. [2] foi adotada aqui para permitir o enfoque dessa análise na eficiência da ED. Ela foi acoplada ao GDE3, substituindo a original. Além disso, foi adotada a mesma codificação real para as variáveis de projeto utilizadas por Silva et al. [4]. As descrições dos problemas e os parâmetros foram idênticos àqueles utilizados por Angelo et al. [2]. Os resultados obtidos mostram que o GDE3 apresentou desempenho superior aos algoritmos MOAS e MOACS e o conjunto de soluções obtido pelo GDE3 contém elementos muito próximos das soluções obtidas por Silva et al. [4], mostrando que o tratamento desses problemas através da versão multiobjetivo não acarreta perda significativa da qualidade das soluções obtidas pela sua versão mono-objetivo.

O restante do texto é organizado da seguinte forma: a Seção 2 faz uma introdução dos conceitos básicos de OE multiobjetivo, enquanto a Seção 3 destaca o algoritmo GDE3. A técnica APM é mostrada na Seção 4 e o algoritmo proposto é descrito em detalhes na Seção 5. Os experimentos numéricos, bem como seus resultados e sua discussão, são mostrados na Seção 6, enquanto a Seção 7 descreve as conclusões do trabalho.

2. Otimização estrutural multiobjetivo

Seja uma treliça de N barras com M graus de liberdade de translação de seus nós, com material de massa específica ρ e comprimento da j -ésima barra denotada por L_j . Considere s_{jl} a tensão normal da j -ésima barra no caso de carregamento l , s_{adm} a tensão normal máxima que esta barra pode estar submetida e u_{il} o deslocamento do nó i no caso de carregamento l . O problema de OE multiobjetivo pode ser escrito como:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_1(\mathbf{x}) = \sum_{j=1}^N \rho A_{[x_j]} L_j \quad \text{e} \quad f_2(\mathbf{x}) = \max(|u_{il}|) \\ \text{s. a.} \quad & |s_{jl}| \leq s_{adm} \\ & \mathbf{x} \in [1, P]^N \\ & j = 1, \dots, N \quad i = 1, \dots, M \quad l = 1, \dots, N_L \end{aligned} \quad (1)$$

Os problemas de OE multiobjetivo tratados neste artigo têm como variáveis de projeto as áreas das seções transversais das barras das treliças, as quais devem ser selecionadas entre os P elementos do conjunto discreto ordenado de forma crescente $AP = \{A_1, \dots, A_P\}$, definido um espaço de busca discreto. Apesar disso, adota-se aqui uma codificação real para as variáveis de projeto conforme sugerido por Kaveh e Shahrrouzi [6] e utilizada por Silva et al. [4]. O vetor $\mathbf{x} = (x_1, \dots, x_N)$ representa a treliça de N barras onde a área da seção transversal da j -ésima barra é $A_{[x_j]}$ ($[x_j]$ -ésimo elemento de AP), onde $[x_j]$ é o inteiro mais próximo de $x_j \in [1, P]$, $j = 1, \dots, N$.

Os objetivos $f_1(\mathbf{x})$ (minimização de custo, aqui representado pelo peso da estrutura) e $f_2(\mathbf{x})$ (minimização dos deslocamentos, aqui representado pelo deslocamento máximo de cada nó) são conflitantes. Isso significa que minimizar os deslocamentos pode acarretar um aumento considerável no peso da estrutura e vice-versa. Assim, a relação de dominância exibida a seguir deve ser observada.

Considerando os vetores $\mathbf{x}, \mathbf{y} \in [1, P]^N$, se $f_1(\mathbf{x}) \leq f_1(\mathbf{y})$ e $f_2(\mathbf{x}) < f_2(\mathbf{y})$, ou equivalentemente, se $f_1(\mathbf{x}) < f_1(\mathbf{y})$ e $f_2(\mathbf{x}) \leq f_2(\mathbf{y})$, então o vetor \mathbf{x} domina o vetor \mathbf{y} . Dado 2 vetores $\mathbf{x}, \mathbf{y} \in [1, P]^N$, uma, e somente uma, das 3 possibilidades ocorre: 1) \mathbf{x} domina \mathbf{y} ; 2) \mathbf{y} domina \mathbf{x} ; 3) \mathbf{x} e \mathbf{y} são não-dominados entre si.

Resolver o Problema 1 através de um AE significa encontrar um conjunto de vetores factíveis $SPO = \{\mathbf{x}_1, \dots, \mathbf{x}_\lambda\}$ onde cada par de elementos de SPO são não-dominados entre si e nenhum outro

vetor factível que domina algum elemento de SPO foi obtido pelo AE. O conjunto das imagens dos elementos de SPO é chamado de Frente de Pareto.

A relação de dominância definida aqui não leva em consideração as violações das restrições dos vetores \mathbf{x} e \mathbf{y} . No algoritmo utilizado neste trabalho, essas violações são tratadas através de penalização de suas funções objetivo, conforme é detalhado na Seção 4. Dessa forma, vetores factíveis e infactíveis podem aparecer na população. Entretanto, como factibilidade é condição necessária para a entrada de um vetor no conjunto SPO , não existem soluções infactíveis nesse conjunto.

3. Generalized Differential Evolution

A ED, proposta originalmente em [7] para problemas mono-objetivos, é sem dúvida um dos mais eficientes AE que utiliza codificação real [8]. Ele se destaca pela sua simplicidade de implementação, incluindo o pequeno número de parâmetros de controle, e pelo seu notável desempenho em comparação com vários outros AE em uma ampla variedade de problemas, inclusive mantendo esse bom desempenho em problemas de grande porte, diferentemente de outros AE que utilizam codificação real [8].

Desde sua proposição, diversos AE baseados em ED para problemas multiobjetivos foram propostos [9]. Adota-se aqui, seguindo a sugestão de Zavala et al. [1], o algoritmo GDE3 proposto por Kukkonen e Lampinen [5].

Nos problemas de OE multiobjetivo de treliças sem restrições, o GDE3 funciona da seguinte maneira:

- Uma população inicial P_0 de POP soluções candidatas é criada aleatoriamente ($P_0 = \{\mathbf{x}_1, \dots, \mathbf{x}_{POP}\}$).
- Cada solução candidata $\mathbf{x} \in P_0$ gera uma nova \mathbf{u} através do **Algoritmo1**, onde $F \in \mathbb{R}$ e $CR \in [0, 1]$ são parâmetros definidos pelo usuário.
- O conjunto R_0 recebe, entre as soluções candidatas \mathbf{x} e \mathbf{u} , aquela que dominar a outra. Caso as 2 sejam não-dominadas entre si, R_0 recebe as 2.

A próxima geração da população P_1 será composta pelas POP melhores soluções candidatas de R_0 , conforme posição na Ordenação de Pareto e valor do indicador de densidade de soluções, *crowding distance* (CD), segundo esquemas propostos por Deb et al. [10].

O esquema de Ordenação de Pareto consiste em classificar o conjunto R_0 em d subconjuntos tais que $F_1 = \{\text{elementos não-dominados de } R_0\}$ (posto 1), $F_2 = \{\text{elementos não-dominados de } R_0 \setminus \{F_1\}\}$ (posto 2), $F_3 = \{\text{elementos não-dominados de } R_0 \setminus \{F_1 \cup F_2\}\}$ (posto 3), ..., $F_d = \{\text{elementos não-dominados de } R_0 \setminus \{F_1 \cup F_2 \cup \dots \cup F_{d-1}\}\}$ (posto d), onde $R_0 = \{F_1 \cup F_2 \cup \dots \cup F_d\}$.

Para o Problema 1, o valor do CD de cada solução candidata é o semi-perímetro do retângulo cujos vértices são as imagens vizinhas mais próximas da sua, dentre todas as imagens das soluções candidatas de mesmo posto. Quanto mais distante a imagem de uma solução candidata se encontra dessas imagens vizinhas, maior é o seu valor de CD (menor densidade de soluções com imagens nesta região). Se uma solução candidata tem uma das funções objetivo como a melhor do seu posto, seu valor de CD é tomado como infinito.

Assim, a próxima geração da população P_1 será composta pelas soluções candidatas de R_0 que estiverem nos subconjuntos com postos menores. Caso exista algum posto d_i tal que $F_1 \cup F_2 \cup \dots \cup F_{d_i}$ tenha mais que POP elementos e

$F_1 \cup F_2 \cup \dots \cup F_{d_i-1}$ tenha menos que POP elementos, deve-se excluir os elementos de F_{d_i} com menores valores de CD até que se tenha $F_1 \cup F_2 \cup \dots \cup F_{d_i}$ com POP elementos. A próxima população será $P_1 = F_1 \cup F_2 \cup \dots \cup F_{d_i}$.

Nesse procedimento, após a exclusão de uma solução candidata de F_{d_i} , todos os valores de CD das restantes são recalculados. Isso pode fornecer uma melhoria na diversidade da população sem a introdução de um novo parâmetro [5].

Algorithm 1.

Geração de uma nova solução candidata \mathbf{u} no GDE3.

- 1: Seja $\mathbf{x} \in$ população.
- 2: Sorteiam-se 3 soluções candidatas $\mathbf{x}_{r1}, \mathbf{x}_{r2}$ e \mathbf{x}_{r3} da população.
- 3: $I_{rand} \leftarrow$ inteiro entre 1 e N escolhido aleatoriamente.
- 4: **para** $i = 1 : N$ **faça**
- 5: $rand \leftarrow$ real entre 0 - 1 escolhido aleatoriamente.
- 6: **se** $rand < CR$ ou $i = I_{rand}$ **então**
- 7: $\mathbf{u}(i) \leftarrow \mathbf{x}_{r1}(i) + F(\mathbf{x}_{r2}(i) - \mathbf{x}_{r3}(i))$.
- 8: **senão**
- 9: $\mathbf{u}(i) \leftarrow \mathbf{x}(i)$.
- 10: **fim se**
- 11: **fim para**

4. Técnica de penalização adaptativa

Para transformar um problema com restrições em um problema sem restrições, tanto Angelo et al. [2] quanto Silva et al. [4], adotaram a APM proposta por Barbosa e Lemonge [3].

Originalmente, a técnica APM foi aplicada em problemas de otimização mono-objetivo com restrições, obtendo resultados competitivos. Entre as suas vantagens estão o tratamento de restrições de igualdade e desigualdade, a não necessidade de que as restrições sejam funções explícitas das variáveis de projeto, a ausência de parâmetros a serem definidos pelo usuário e a facilidade de implementação computacional.

Apesar da sua proposição inicial ser para problemas mono-objetivos, a técnica APM pode ser adaptada para problemas multiobjetivos facilmente, apenas repetindo seu procedimento para cada função objetivo separadamente, como foi feito por Angelo et al. [2].

Para a função objetivo f_1 da Equação 1, a técnica APM funciona da seguinte maneira: conhecidos os valores para f_1 de todas as POP soluções candidatas de P_G (população na geração G) e supondo N_c o número total de restrições, calcula-se para cada $\mathbf{x} \in P_G$ os valores de $F_1(\mathbf{x})$ pela fórmula

$$F_1(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}), & \text{se } \mathbf{x} \text{ efactível} \\ \bar{f}_1(\mathbf{x}) + \sum_{m=1}^{N_c} k_m \delta_m(\mathbf{x}), & \text{caso contrário} \end{cases} \quad (2)$$

onde

$$\bar{f}_1(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}), & \text{se } f_1(\mathbf{x}) > \langle f_1(\mathbf{x}) \rangle \\ \langle f_1(\mathbf{x}) \rangle, & \text{caso contrário} \end{cases} \quad (3)$$

sendo $\bar{f}_1(\mathbf{x})$ a média dos valores de $f_1(\mathbf{x})$ de todas as POP soluções candidatas.

A função δ_m é dada por $\delta_m(\mathbf{x}) = \max(|s_m| - s_{adm}, 0)$ e o coeficiente de penalização k_j da j -ésima restrição é definido por

$$k_j = |f_1(\mathbf{x})| \frac{(\delta_j(\mathbf{x}))}{\sum_{m=1}^{N_G} [(\delta_m(\mathbf{x}))]^2} \quad (4)$$

Esse procedimento se repete para a segunda função objetivo, obtendo-se $F_2(\mathbf{x})$ para cada $\mathbf{x} \in P_G$ a partir dos valores de f_2 . O APM foi projetado de tal forma que os valores dos coeficientes de penalização correspondentes às restrições mais difíceis de serem atendidas sejam maiores.

5. Algoritmo proposto

A proposta do algoritmo GDE3 acoplado à técnica APM usada nos experimentos numéricos consiste em substituir o uso de f_1 e f_2 no esquema de seleção da nova geração pelo uso dos valores de F_1 e F_2 . Isto significa que a avaliação da dominância do vetor \mathbf{u} sobre o vetor \mathbf{x} , o processo de Ordenação de Pareto do conjunto R_t e o cálculo dos valores de CD de cada posto foram realizados considerando os valores de F_1 e F_2 . Além disso, como em Angelo et al. [2] foi definido que o SPO de cada algoritmo corresponde a todas as soluções candidatas não-dominadas factíveis obtidas ao longo de toda sua execução, isso também foi adotado aqui. O detalhamento da proposta pode ser visto no **Algoritmo2**.

Algorithm 2.

GDE3+APM.

- 1: Defina os parâmetros CR , F e $GEN \triangleright GEN$ é o número máximo de gerações.
- 2: $P_0 \leftarrow$ população inicial criada aleatoriamente de POP soluções candidatas $\mathbf{x} \in [1, P]^N$.
- 3: $Arq \leftarrow P_0 \triangleright$ Arquivo externo que vai armazenar todas as soluções candidatas encontradas durante toda a execução do algoritmo.
- 4: $f_1, f_2 \leftarrow$ funções objetivo de cada solução candidata de P_0 .
- 5: $G \leftarrow 0$ contador de gerações.
- 6: **enquanto** $G \leq GEN$ **faça**
- 7: $F_1, F_2 \leftarrow$ funções objetivo de cada solução candidata de P_G modificadas pela técnica APM.
- 8: **para** $i = 1 : POP$ **faça**
- 9: $\mathbf{u} \leftarrow$ criado a partir de $\mathbf{x}_i \in P_G$ através do **Algoritmo1**.
- 10: Obtenha $f_1(\mathbf{u})$ e $f_2(\mathbf{u})$.
- 11: Obtenha $F_1(\mathbf{u})$ e $F_2(\mathbf{u})$ modificadas pela técnica APM com os mesmos valores dos coeficientes de penalização usados na linha 7.
- 12: **se** \mathbf{u} domina \mathbf{x}_i **então**
- 13: $R_G \leftarrow \mathbf{u}$.
- 14: **senão**
- 15: **se** \mathbf{x}_i domina \mathbf{u} **então**
- 16: $R_G \leftarrow \mathbf{x}_i$.
- 17: **senão**
- 18: $R_G \leftarrow \mathbf{x}_i, \mathbf{u}$.
- 19: **fim se**

20: **fim se**

• 21: **fim para**

• 22: **se** R_G tem mais de POP soluções candidatas **então**

• 23: Classifique R_G em d postos pelo processo de Ordenação de Pareto.

• 24: $P_{G+1} \leftarrow \emptyset$.

• 25: $\gamma \leftarrow 1$.

• 26: **enquanto** Quantidade de soluções candidatas em $P_{G+1} < POP$ **faça**

• 27: $P_{G+1} \leftarrow P_{G+1} \cup$ soluções candidatas do posto γ .

• 28: $\gamma \leftarrow \gamma + 1$.

• 29: **fim enquanto**

• 30: **enquanto** Quantidade de soluções candidatas em $P_{G+1} > POP$ **faça**

• 31: Calcule CD para todas as soluções candidatas de posto $\gamma - 1$.

• 32: Exclua de P_{G+1} a solução candidata de posto $\gamma - 1$ com menor valor de CD .

• 33: **fim enquanto**

• 34: **senão**

• 35: $P_{G+1} \leftarrow R_G$.

• 36: **fim se**

• 37: $Arq \leftarrow Arq \cup P_{G+1}$

• 38: **fim enquanto**

• 39: $SPO \leftarrow$ soluções candidatas não-dominadas e factíveis de Arq .

• 40: Retorne SPO .

6. Experimentos numéricos

As descrições dos problemas, os parâmetros e as métricas de desempenho adotadas aqui são idênticas às utilizadas por Angelo et al. [2]. Cada algoritmo foi executado 100 vezes com tamanho da população de 50 soluções candidatas e 1.000 gerações, perfazendo um total de 50.000 avaliações de cada uma das funções objetivo. Da mesma forma, os parâmetros do GDE3 adotados em todos os problemas são os mesmos utilizados por Kukkonen e Lampinen [11] para problemas com restrições: $CR = 0, 4$ e $F = 0, 3$.

As métricas de desempenho adotadas foram o Hipervolume, proposta por Zitzler e Thiele [12], e a métrica *Empirical Attainment Function*, proposta por Fonseca e Fleming [13].

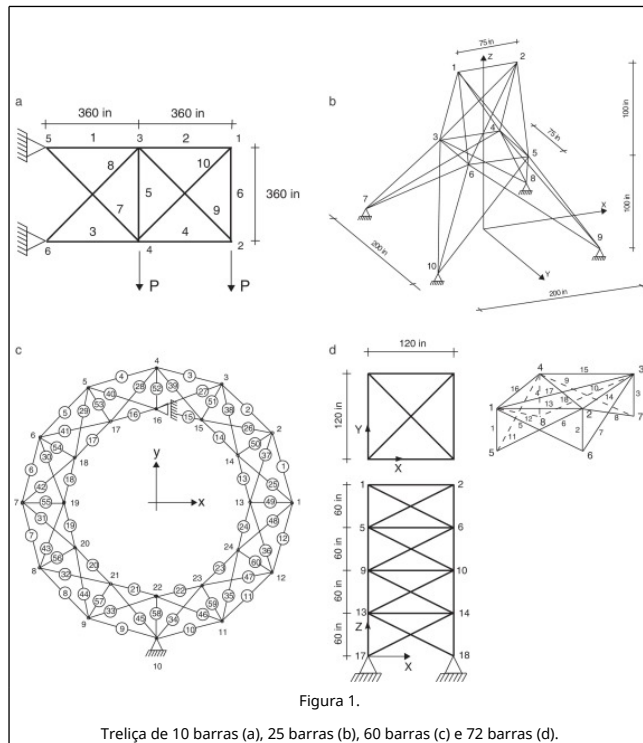
No caso de 2 funções objetivo, a métrica Hipervolume fornece o valor de H (área do polígono limitado pela Frente de Pareto da SPO e um dado ponto de referência). O algoritmo que gerou a SPO que obteve o maior valor de H será o melhor.

A métrica *Empirical Attainment Function* (EAF) atua como uma distribuição de probabilidade. Dada uma probabilidade $\alpha\%$, com $0 < \alpha < 100$, o EAF é um conjunto de pontos não-dominados entre si de tal forma que qualquer solução arbitrária desse algoritmo obtida por qualquer execução tem $100 - \alpha\%$ de chance de dominar pelo menos um desses pontos. Quando $\alpha = 0$, o EAF é um conjunto de pontos não-dominados entre si de tal forma que qualquer solução arbitrária desse algoritmo obtida por qualquer execução tem 0% de chance de ser dominada por algum desses pontos. Já quando $\alpha = 100$, o EAF é o conjunto de todos os pontos não-dominados entre si obtidos em todas as execuções. O algoritmo cujo EAF tem maior hipervolume é o melhor algoritmo. Uma discussão mais detalhada sobre a obtenção do EAF pode ser encontrada em [13].

6.1. Problemas

6.1.1. Treliça de 10 barras

O primeiro problema analisado é a treliça clássica de 10 barras, ilustrado pela figura 1a. A versão mono-objetivo é amplamente estudada [14] and [4] e trabalhos sobre sua versão multiobjetivo podem ser encontrados em [15], [16], [17], [18] and [2].



A densidade do material é $\rho = 0, 1lb/in^3$, a tensão normal máxima é limitada em $\pm 25 ksi$, o Módulo de Young $E = 10^4 ksi$ e cargas de 100 kips são aplicadas nos nós 2 e 4 na direção y (representadas na figura 1a por P).

O conjunto das áreas possíveis é (em in^2): $AP = \{1,62; 1,80; 1,99; 2,13; 2,38; 2,62; 2,63; 2,88; 2,93; 3,09; 3,13; 3,38; 3,47; 3,55; 3,63; 3,84; 3,87; 3,88; 4,18; 4,22; 4,49; 4,59; 4,80; 4,97; 5,12; 5,74; 7,22; 7,97; 11,50; 13,50; 13,90; 14,20; 15,50; 16,00; 16,90; 18,80; 19,90; 22,00; 22,90; 26,50; 30,00; 33,50\}$, definindo $P = 42$.

6.1.2. Treliça de 25 barras

O segundo problema analisado é a treliça de 25 barras, ilustrado pela figura 1b. Da mesma forma que a treliça de 10 barras, a versão mono-objetivo também é amplamente estudada [14] and [4] e trabalhos sobre sua versão multiobjetivo podem ser encontrados em [19], [17], [20], [21] and [2].

A densidade do material é $\rho = 0, 1lb/in^3$, a tensão normal máxima é limitada em $\pm 40 ksi$ e o Módulo de Young $E = 10^4 ksi$. O conjunto das áreas possíveis é (em in^2): $AP = \{0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1,0; 1,1; 1,2; 1,3; 1,4; 1,5; 1,6; 1,7; 1,8; 1,9; 2,0; 2,1; 2,2; 2,3; 2,4; 2,5; 2,6; 2,8; 3,0; 3,2; 3,4\}$, definindo $P = 30$.

As barras da treliça são agrupadas e cada grupo possui uma única área A, a fim de manter a simetria da estrutura. O

agrupamento das barras é mostrado na tabela 1 e o carregamento aplicado sobre a estrutura é mostrado na tabela 2.

Tabela 1. Agrupamento para a treliça de 25 barras.

Grupo	Conectividade
A_1	1-2
A_2	1-4, 2-3, 1-5, 2-6
A_3	2-5, 2-4, 1-3, 1-6
A_4	3-6, 4-5
A_5	3-4, 5-6
A_6	3-10, 6-7, 4-9, 5-8
A_7	3-8, 4-7, 6-9, 5-10
A_8	3-7, 4-8, 5-9, 6-10

Tabela 2. Carregamento para a treliça de 25 barras (em kips).

Nó	F_x	F_y	F_z
1	1	-10,0	-10,0
2	0	-10,0	-10,0
3	0,5	0	0
6	0,6	0	0

6.1.3. Treliça de 60 barras

O terceiro problema analisado foi o da treliça de 60 barras em formato de anel, proposto em [22] e ilustrado pela figura 1c. A versão mono-objetivo pode ser encontrada em [23] and [4] e somente em [2] foi encontrada a versão multiobjetivo desse problema.

A densidade do material é $\rho = 0, 1lb/in^3$, a tensão normal máxima é limitada em $\pm 10 ksi$ e o Módulo de Young $E = 10^4 ksi$. O raio externo do anel dessa treliça mede 100 in e o raio interno 90 in. O conjunto das áreas possíveis é (em in^2): $AP = \{0,5; 0,6; 0,7; \dots; 4,7; 4,8; 4,9\}$, definindo $P = 45$.

A treliça é submetida a 3 casos de carregamento, observados na tabela 4, totalizando 180 restrições. As barras são agrupadas conforme dados da tabela 3.

Tabela 4. Carregamento para a treliça de 60 barras (em kips).

Carregamento	Nó	F_x	F_y
1	1	-10,0	0
	7	9,0	0
2	15	-8,0	3,0
	18	-8,0	3,0
3	22	-20,0	10,0

Tabela 3. Agrupamento para a treliça de 60 barras.

Grupo	Barras	Grupo	Barras
A_1	49 ao 60	A_{14}	25 e 37
A_2	1 e 13	A_{15}	26 e 38

A_3	2 e 14	A_{16}	27 e 39
A_4	3 e 15	A_{17}	28 e 40
A_5	4 e 16	A_{18}	29 e 41
A_6	5 e 17	A_{19}	30 e 42
A_7	6 e 18	A_{20}	31 e 43
A_8	7 e 19	A_{21}	32 e 44
A_9	8 e 20	A_{22}	33 e 45
A_{10}	9 e 21	A_{23}	34 e 46
A_{11}	10 e 22	A_{24}	35 e 47
A_{12}	11 e 23	A_{25}	36 e 48
A_{13}	12 e 24		

6.1.4. Treliça de 72 barras

O quarto problema analisado foi o da treliça de 72 barras, ilustrado pela figura 1d. A versão mono-objetivo pode ser encontrada em [14], [23] and [4] e a versão multiobjetivo foi encontrada somente em [2].

A densidade do material é $\rho = 0, 1lb/in^3$, a tensão normal máxima é limitada em $\pm 25 ksi$ e o Módulo de Young $E = 10^4 ksi$. O conjunto das áreas possíveis é (em in^2): $AP = \{0,1; 0,2; 0,3; \dots; 2,4; 2,5\}$, definindo $P = 25$.

A treliça é submetida a 2 casos de carregamento, observados na tabela 6, totalizando 60 restrições de tensão. As barras são agrupadas conforme tabela 5.

Tabela 6. Carregamento para a treliça de 72 barras (em kips).

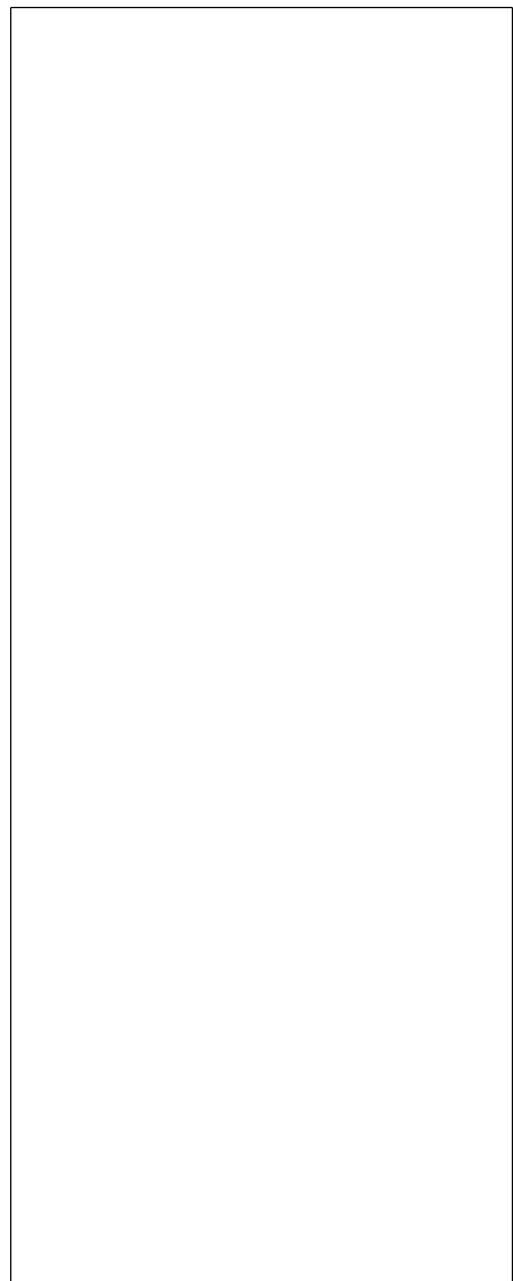
Carregamento	Nó	F_x	F_y	F_z
1	1	5	5	-5
2	1	0	0	-5
	2	0	0	-5
	3	0	0	-5
	4	0	0	-5

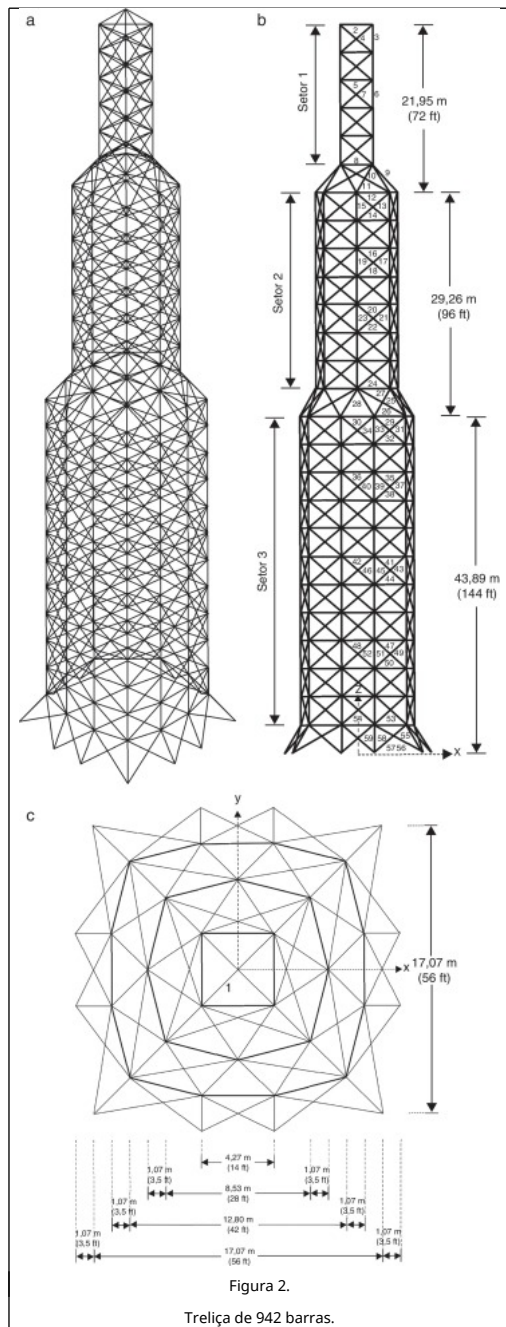
Tabela 5. Agrupamento para a treliça de 72 barras.

Grupo	Barras
A_1	1, 2, 3 e 4
A_2	5, 6, 7, 8, 9, 10, 11 e 12
A_3	13, 14, 15 e 16
A_4	17 e 18
A_5	19, 20, 21 e 22
A_6	23, 24, 25, 26, 27, 28, 29 e 30
A_7	31, 32, 33 e 34
A_8	35 e 36
A_9	37, 38, 39 e 40
A_{10}	41, 42, 43, 44, 45, 46, 47 e 48
A_{11}	49, 50, 51 e 52
A_{12}	53 e 54
A_{13}	55, 56, 57 e 58
A_{14}	59, 60, 61, 62, 63, 64, 65 e 66
A_{15}	67, 68, 69 e 70
A_{16}	71 e 72

6.1.5. Treliça de 942 barras

O último problema analisado aqui foi a treliça de 942 barras, ilustrada pela figura 2. As 942 barras são agrupadas em 59 variáveis de projeto e considera-se um caso de carregamento que consiste em cargas horizontais e verticais da seguinte forma: cargas verticais na direção z de $-3 kips$, $-6 kips$ e $-9 kips$ em cada nó no primeiro, segundo e terceiro setores, respectivamente; cargas laterais na direção y de $1 kips$ em todos os nós da torre; e cargas laterais na direção x de $1, 5 kips$ e $1 kips$ em cada nó à esquerda e à direita da torre, respectivamente. As variáveis de projeto são valores inteiros no intervalo de $[1, 200]in^2$ e as restrições incluem tensão normal máxima limitada em $\pm 25 ksi$ para todas as barras. O Módulo de Young é $E = 10^4 ksi$ e a densidade do material é de $\rho = 0, 1lb/in^3$. A versão mono-objetivo pode ser encontrada em [24], [25], [26] and [27] e a versão multiobjetivo foi encontrada somente em [2].





6.2. Discussão dos resultados

A tabela 7 mostra as médias e os desvios padrão (DP) dos resultados da métrica Hipervolume, calculados para cada uma das 100 execuções realizadas. Para o cálculo de H , optou-se pela normalização de todos os SPO , mapeando cada função objetivo no intervalo de $[0, 1]$ de tal forma que o valor mínimo correspondesse ao 0 e o valor máximo ao 1. O ponto de referência é dado pelos melhores valores extremos de todos os SPO gerados por todos os algoritmos em um dado problema, o qual normalizado fica $[1, 1]$.

Tabela 7. Média e Desvio Padrão (DP) dos Resultados de H .

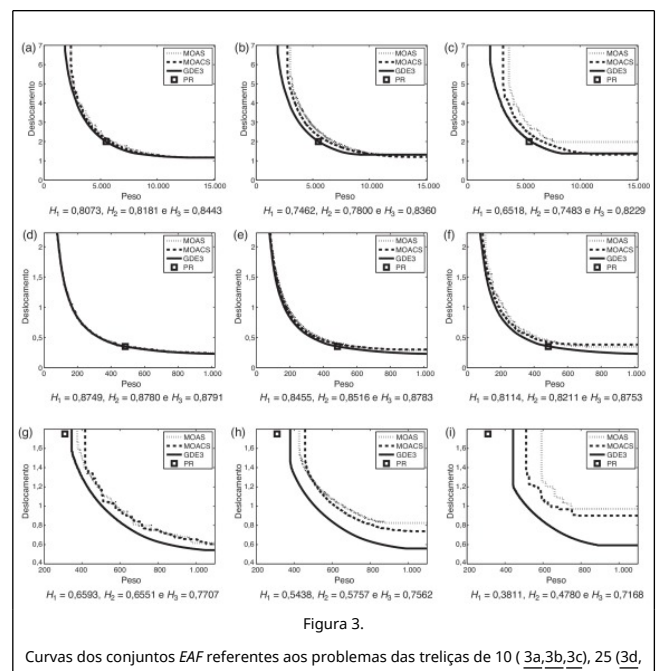
		GDE3	MOACS	MOAS
10 barras	Média	8,481641	$7,977419 \times 10^{-1}$	$7,574947 \times 10^{-1}$

		$\times 10^{-1}$	(+)	(+)
25 barras	DP	$3,225481 \times 10^{-1}$	$4,470697 \times 10^{-3}$	$1,857184 \times 10^{-2}$
	Média	$8,854238 \times 10^{-1}$	$8,585920 \times 10^{-1}$ (+)	$8,526323 \times 10^{-1}$ (+)
60 barras	DP	$7,173082 \times 10^{-1}$	$6,053763 \times 10^{-3}$	$4,116940 \times 10^{-3}$
	Média	$7,536069 \times 10^{-1}$	$5,721502 \times 10^{-1}$ (+)	$5,391532 \times 10^{-1}$ (+)
72 barras	DP	$7,866109 \times 10^{-1}$	$1,723658 \times 10^{-2}$	$2,870502 \times 10^{-2}$
	Média	$9,179434 \times 10^{-1}$	$8,586737 \times 10^{-1}$ (+)	$8,569383 \times 10^{-1}$ (+)
942 barras	DP	$2,003992 \times 10^{-1}$	$7,142308 \times 10^{-3}$	$4,078346 \times 10^{-3}$
	Média	$9,148664 \times 10^{-1}$	$7,304951 \times 10^{-1}$ (+)	$7,244967 \times 10^{-1}$ (+)
	DP	$1,130059 \times 10^{-1}$	$1,614781 \times 10^{-2}$	$6,067129 \times 10^{-2}$

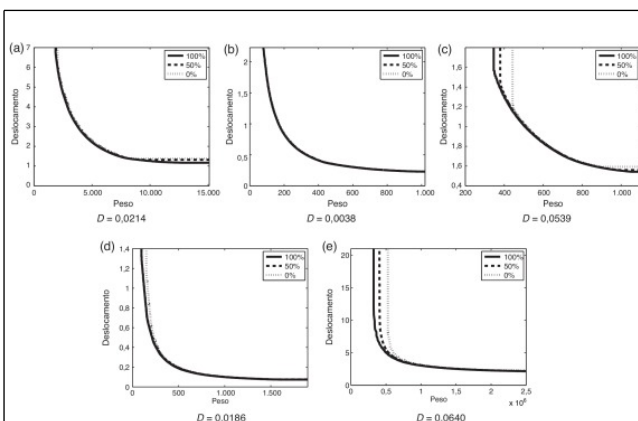
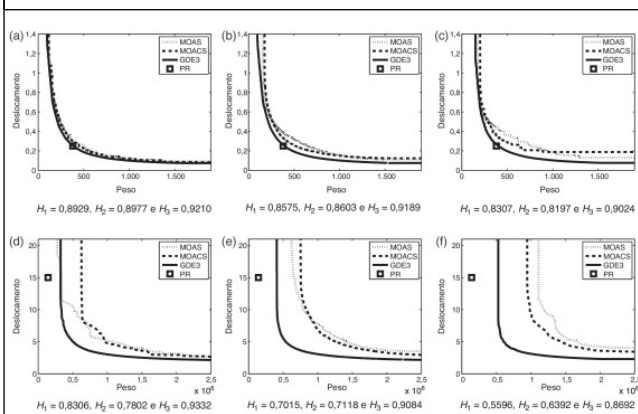
O sinal (+) indica que a diferença entre os 100 resultados obtidos naquela métrica é estatisticamente significativa (p -valor $\leq 0,05$), conforme o teste não-paramétrico de Wilcoxon calculado através da função *ranksum* do MATLAB®. Para cada problema, o melhor resultado em cada métrica está marcado em negrito.

Pelos resultados exibidos na tabela 7 pode-se observar que o desempenho do GDE3 é melhor do que o dos algoritmos MOAS e MOACS na métrica Hipervolume. A média dos valores de H dos SPO s obtidos pelo GDE3 é maior do que a média dos valores de H dos SPO s obtidos pelo MOAS e pelo MOACS em todos os problemas, com diferença estatisticamente significativa. Além disso, o desvio padrão dos 100 valores de H obtidos pelo GDE3 é o menor.

Em relação à métrica *Empirical Attainment Function*, foi definida as probabilidades de 100, 50 e 0%. Os conjuntos EAFs foram gerados pelo pacote EAF Tools¹. Os gráficos das figuras 3, 4 e 5 referem-se às curvas formadas pelos pontos dos conjuntos EAF para os problemas adotados. Os valores dos hipervolumes normalizados de cada uma dessas curvas estão indicadas na própria figura por H_1 , H_2 e H_3 referindo-se aos algoritmos MOAS, MOACS e GDE3, respectivamente.



3e,3f) e 60 (3g,3h,3i) barras com probabilidades de 100% (3a,3d,3g), 50% (3b,3e,3h) e 0% (3c,3f,3i). Os valores dos hipervolumes normalizados de cada curva dos algoritmos MOAS, MOACS e GDE3 estão indicadas por H_1 , H_2 e H_3 , respectivamente.



Esses gráficos também apresentam a imagem da melhor solução do problema mono-objetivo como um ponto de referência ($f_1; f_2$). São eles (5.490,70; 2) para a treliça de 10 barras, (484, 85; 0, 35) para a treliça de 25 barras, (309, 44; 1, 75) para a treliça de 60 barras e (379, 667; 0, 25) para a treliça de 72 barras, todos encontrados em [4]. O ponto de referência para a treliça de 942 barras é (142.295,75; 15), solução da versão mono-objetivo do problema encontrado em [26]. Isso mostra que a qualidade das soluções é mantida na versão multiobjetivo, visto que a Frente de Pareto se encontra bastante próxima desses pontos de referência.

Observando-se os gráficos das figuras 3, 4 e 5, percebe-se que a curva do conjunto EAF do algoritmo GDE3 domina, no geral, uma área muito maior do que os demais em todos os problemas e em todas as probabilidades. Nos problemas da treliça de 10 e 942 barras, o algoritmo GDE3 apresenta apenas uma parte dessa curva ligeiramente dominada por pelo menos um dos outros algoritmos. Nos demais problemas, os algoritmos MOAS e MOACS apresentam, no máximo, pequenas partes da curva coincidentes com a do algoritmo GDE3.

No problema da treliça de 60 barras, as curvas dos algoritmos

MOAS e MOACS ficaram totalmente dominadas pela do algoritmo GDE3 em todas as probabilidades. Em todos os problemas foram encontradas soluções cujas imagens são muito próximas ao ponto de referência pelo algoritmo GDE3. Além disso, as imagens das soluções obtidas pelo algoritmo GDE3 estão mais próximas do ponto de referência do que as imagens das soluções obtidas pelos algoritmos MOAS e MOACS.

Os gráficos das 3 curvas dos conjuntos EAF do algoritmo GDE3 apresentado na figura indicam ainda que ele é mais estável do que os demais, uma vez que a área entre as curvas relativas às probabilidades de 0 e 100% é bem menor quando comparada aos algoritmos MOAS e MOACS. Essa diferença está indicada por D em cada figura, que é dado pela diferença entre os valores dos hipervolumes normalizados das curvas correspondentes às probabilidades de 100 e 0%. Além disso, as curvas do algoritmo GDE3 relativas às probabilidades de 50 e 100% são muito próximas em todos os problemas, indicando que a maioria das soluções obtidas por ele tem imagens próximas das melhores.

7. Conclusões

Neste trabalho foi analisado o desempenho do algoritmo GDE3 combinado com a APM em 5 problemas de OE de treliças com codificação real para as variáveis de projeto, as quais pertencem a um espaço de busca discreto. Esses problemas foram resolvidos por Angelo et al. [2] utilizando 2 algoritmos baseados em colônia de formigas: MOAS e MOACS. Compararam-se aqui os resultados obtidos pelos algoritmos GDE3, MOAS e MOACS em 2 métricas: Hipervolume e *Empirical Attainment Function*.

A versão mono-objetivo de 4 desses problemas foram resolvidos via ED por Silva et al. [4] também com codificação real para as variáveis de projeto. A mesma técnica de tratamento de restrições APM foi utilizada em [2] e [4].

A comparação dos resultados indica que o GDE3 apresentou desempenho significativamente superior aos demais algoritmos nas 2 métricas. Na métrica Hipervolume, a média dos valores de H dos SPO obtidos pelo GDE3 é maior do que a média dos valores de H dos SPO obtidos pelo MOAS e MOACS em todos os problemas, com diferença estatisticamente significante entre todos os resultados. Além disso, um menor desvio padrão dos valores de H relativos ao GDE3 foi observado em todos os problemas quando comparado ao desvio padrão dos valores de H relativos ao MOAS e ao MOACS.

Na métrica *Empirical Attainment Function*, o conjunto EAF do algoritmo GDE3 domina, em geral, uma área bem maior do que a dominada pelos conjuntos EAF dos algoritmos MOAS e MOACS em todos os problemas e em todas as probabilidades consideradas aqui (0, 50 e 100%). A área entre as curvas EAF relativas às probabilidades de 0 e 100% do algoritmo GDE3 é bem menor quando comparada à área entre as curvas EAF relativas às mesmas probabilidades dos algoritmos MOAS e MOACS. Em alguns casos, as curvas EAF relativas às probabilidades de 0 e 100% do algoritmo GDE3 foram quase coincidentes, indicando que ele é o algoritmo mais estável entre todos os analisados.

Também observou-se nos experimentos que em todos os problemas foram encontradas soluções cujas imagens são muito próximas ao ponto de referência pelo algoritmo GDE3, sendo essas mais próximas do ponto de referência do que as obtidas pelos algoritmos MOAS e MOACS. Isso indica que não há perda significativa da qualidade das soluções obtidas pelo GDE3 em relação aos resultados da versão mono-objetivo desses problemas.

Como Zavala et al. [1] também aponta a carência de trabalhos

envolvendo a incorporação de preferências do usuário nos AE aplicados a OE multiobjetivo, para trabalhos futuros está previsto experimentos com essa característica. Isso pode permitir a busca por soluções cujas imagens estão em uma região específica da Frente de Pareto, melhorando a qualidade das soluções a serem apresentadas ao(s) tomador(es) de decisão.

Agradecimentos

Os autores agradecem às agências de fomento pelos seguintes apoios CNPq (306815/2011- 7) e FAPEMIG (TEC PPM 528/11 e TEC PPM 388/14).

References

- [1] G. Zavala, A.J. Nebro, F. Luna, C. Coello; A survey of multi-objective metaheuristics applied to structural optimization; *Structural and Multidisciplinary Optimization*, 49 (4) (2014), pp. 537-558
- [2] J.S. Angelo, H.S. Bernardino, H.J.C. Barbosa; Multi-objective ant colony approaches for structural optimization problems; *Proceedings of the Eleventh International Conference on Computational Structures Technology* (2012), p. 66
- [3] H.J.C. Barbosa, A.C.C. Lemonge; An adaptive penalty scheme in genetic algorithms for constrained optimization problems; *GECCO'02: Proceedings of the Genetic and Evolutionary Computation Conference* (2002), pp. 287-294
- [4] E.K. Silva, H.J.C. Barbosa, A.C.C. Lemonge; An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization; *Optimization and Engineering*, 12 (1-2) (2011), pp. 31-54
- [5] S. Kukkonen, J. Lampinen; Gde3: The third evolution step of generalized differential evolution; *IEEE Congress on Evolutionary Computation (CEC'2005)*, IEEE (2005), pp. 443-450
- [6] A. Kaveh, M. Shahrouzi; Direct index coding for discrete sizing optimization of structures by genetic algorithms; *International Journal of Civil Engineering*, 3 (3-4) (2005), pp. 166-181
- [7] R. Storn, K. Price; Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces; *Journal of Global Optimization*, 11 (1997), pp. 341-359
- [8] S. Das, P. Suganthan; Differential evolution: A survey of the state-of-the-art, *Evolutionary Computation*; *IEEE Transactions on*, 15 (1) (2011), pp. 4-31 doi:10.1109/TEVC.2010.2059031
- [9] E. Mezura-Montes, M. Reyes-Sierra, C.A.C. Coello; Multi-objective optimization using differential evolution: A survey of the state-of-the-art; U. Chakraborty (Ed.), *Advances in Differential Evolution*, Springer Berlin Heidelberg (2008), pp. 173-196
- [10] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan; A fast and elitist multiobjective genetic algorithm: Nsga-ii; *IEEE Transactions on Evolutionary Computation*, 6 (2) (2002), pp. 182-197
- [11] S. Kukkonen; Generalized differential evolution for global multi-objective optimization with constraints, Ph. D. thesis; *Lappeenranta University of Technology, Finlândia* (2012)
- [12] E. Zitzler, L. Thiele; Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach.; *IEEE Transactions on Evolutionary Computation*, 3 (4) (1999), pp. 257-271
- [13] C. Fonseca, P. Fleming; On the performance assessment and comparison of stochastic multiobjective optimizers; *Proceedings of Parallel Problem Solving from Nature IV (PPSN-IV)* (1996), pp. 584-593
- [14] A.C.C. Lemonge, H.J.C. Barbosa; An adaptive penalty scheme for genetic algorithms in structural optimization; *International Journal For Numerical Methods In Engineering*, 59 (5) (2004), pp. 703-706
- [15] P. Hajela, C. Lin; Genetic search strategies in multicriterion optimal design; *Struct Multidiscip Optim*, 4 (2) (1992), pp. 99-107
- [16] D. Buche, R. Dornberger; New evolutionary algorithm for multi-objective optimization and its application to engineering design problems; in: *Fourth world congress of structural and multidisciplinary optimization* (2001)
- [17] G. Luh, C. Chueh; Multi-objective optimal design of truss structure with immune algorithm; *Comput Struct*, 82 (11-12) (2004), pp. 829-844
- [18] R. Su, X. Wang, L. Gui, Z. Fan; Multi-objective topology and sizing optimization of truss structures based on adaptive multi-island search strategy; *Struct Multidiscip Optim*, 43 (2) (2010), pp. 275-286
- [19] C. Coello, A. Christiansen; Multiobjective optimization of trusses using genetic algorithms; *Comput. Struct.*, 75 (6) (2000), pp. 647-660
- [20] K. Izui, S. Nishiwaki, M. Yoshimura, M. Nakamura, J. Renaud; Enhanced multiobjective particle swarm optimization in combination with adaptive weighted gradient-based searching; *Eng. Optim.*, 40 (9) (2008), pp. 789-804
- [21] A. Kaveh, K. Laknejadi; A hybrid multi-objective optimization and decision making procedure for optimal design of truss structures; *IJST Trans. Civ. Eng.*, 35 (C2) (2011), pp. 137-154
- [22] S. Patnaik, D. Hopkins, R. Coroneos; Structural optimization with approximate sensitivities; *Computer and Structures*, 58 (2) (1996), pp. 407-418
- [23] H. Bernadino; Hybridization of genetic algorithms and artificial immune systems for constrained optimization problems in engineering (in portuguese), Ph.D. thesis; *Programa de Pós-Graduação em Modelagem Computacional, Universidade Federal de Juiz de Fora, Brasil* (2008)
- [24] O. Hasançebi; Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures; *Computers & Structures*, 86 (1-2) (2008), pp. 119-132
- [25] H. Adeli, N. Cheng; Concurrent genetic algorithms for optimization of large structures; *Journal of Aerospace Engineering*, 7 (3) (1994), pp. 276-296
- [26] F. Erbatur, O. Hasançebi, I. Tütüncü, H. Kiliç; Optimal design of planar and space structures with genetic algorithms; *Computers & Structures*, 75 (2) (2000), pp. 209-224
- [27] O. Hasançebi, F. Erbatur; On efficient use of simulated annealing in complex structural optimization problems; *Acta Mechanica*, 157 (2002), pp. 27-50

Notes

1. Disponível em <http://iridia.ulb.ac.be/manuel/eaftools>